Beyond RGB: Scene-Property Synthesis with Neural Radiance Fields

Mingtong Zhang*,1 Shuhong Zheng*,1 Zhipeng Bao[2] Martial Hebert[2] Yu-Xiong Wang[1]
[1]University of Illinois Urbana-Champaign [2]Carnegie Mellon University
{mz62, szheng36, yxw}@illinois.edu {zbao, hebert}@cs.cmu.edu

Abstract

Comprehensive 3D scene understanding, both geometrically and semantically, is important for real-world applications such as robot perception. Most of the existing work has focused on developing data-driven discriminative models for scene understanding. This paper provides a new approach to scene understanding, from a synthesis model perspective, by leveraging the recent progress on implicit scene representation and neural rendering. Building upon the great success of Neural Radiance Fields (NeRFs), we introduce Scene-Property Synthesis with NeRF (SS-NeRF) that is able to not only render photo-realistic RGB images from novel viewpoints, but also render various accurate scene properties (e.g., appearance, geometry, and semantics). By doing so, we facilitate addressing a variety of scene understanding tasks under a unified framework, including semantic segmentation, surface normal estimation, reshading, keypoint detection, and edge detection. Our SS-NeRF framework can be a powerful tool for bridging generative learning and discriminative learning, and thus be beneficial to the investigation of a wide range of interesting problems, such as studying task relationships within a synthesis paradigm, transferring knowledge to novel tasks, facilitating downstream discriminative tasks as ways of data augmentation, and serving as auto-labeller for data creation. Our code is available at https://github.com/zsh2000/SS-NeRF.

1. Introduction

Consider a domestic robot that is navigating in a room and performing various types of household tasks. To do so, the robot needs a comprehensive geometric and semantic understanding of the scene, uncovering the complete 3D spatial layout, functional attributes, and semantic labels of the scene, *etc*. [39]. Most of the existing work on 3D scene understanding has focused on developing data-driven *discriminative* models for various scene analysis problems [23, 27], such as semantic segmentation, object detection, and surface normal estimation. By contrast, this paper introduces a novel perspective for scene understanding – instead of developing discriminative models, we learn an expressive 3D scene representation through the process of synthesizing various paired scene properties with neural rendering.

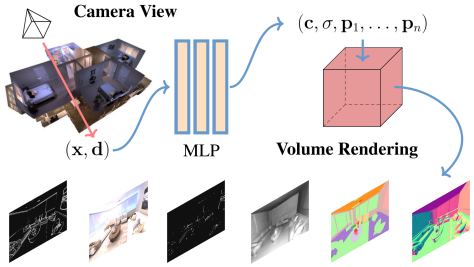

Figure 1: We represent the scene as an implicit function and develop a *versatile* neural scene representation, SS-NeRF, that is able to not only render images from novel viewpoints, but also render various scene properties (*e.g.*, appearance, geometry, and semantics) paired with the synthesized images, under a unified framework.

An important first step toward synthesizing various scene properties is rendering photo-realistic images. One of the most influential recent advances in this direction is Neural Radiance Field (NeRF) [37] which, given a handful of images of a static scene, learns an implicit volumetric representation of the scene that can be rendered from novel viewpoints. By sampling the coordinates along each camera ray from various views, NeRF represents a complex scene as a continuous 5D implicit function with a multilayer perceptron network, which regresses from a single 5D coordinate to a single volume density and view-dependent RGB color. In the end, NeRF accumulates those colors and densities into a 2D image through volume rendering. The implicit representation is optimized by minimizing the residual between synthesized images and ground-truth from various views. NeRF has inspired significant follow-up work that has primarily focused on improving the quality of rendered images [42, 50], speeding up the training and rendering [12, 17, 33, 47], *etc*.

In this paper, we are interested in a different question: *Could this implicit representation be extended to synthesize richer scene properties beyond RGB color?* The answer is **yes**. A well-developed method for generative models [2] is to train a NeRF model, and then predict different scene

*Equal contribution

properties with discriminative models. However, this hybrid solution contains a natural gap between the synthesis model and the discriminative model. To better bridge generative learning and discriminative learning, as illustrated in Fig. 1, we develop a NeRF-style model that is able to render not only photo-realistic RGB images from novel viewpoints, but also various accurate scene properties corresponding to the synthesized images, *under a unified framework*. This thus facilitates comprehensive scene understanding including semantic segmentation, surface normal estimation, reshading, keypoint detection, and edge detection. We call our framework *Scene-Property Synthesis with NeRF* (SS-NeRF).

Naturally, we find that some of the scene properties are sensitive to the observation directions, while others are not (*e.g.*, semantic labels), for which the view direction input of the original NeRF model is redundant. Therefore, we adopt two branches to take care of these different properties (shown in Fig. 2) that consider or ignore the view direction input (θ, ϕ) respectively. By doing so, the proposed SS-NeRF model is able to deal with different types of properties in a coherent way, yielding realistic synthesis for all of them. Moreover, the learned scene representation is shareable and beneficial across different properties, leading to the result that SS-NeRF is able to generalize from synthesizing a single property to multiple properties.

As a general, flexible framework, SS-NeRF further facilitates the investigation of a variety of interesting problems. For example, within the SS-NeRF framework, we analyze the relationship among different scene properties through both multi-task learning and knowledge transfer. We show that a learned implicit geometric and semantic representation enables the flow of knowledge across different synthesis tasks, so that they can benefit one another. While similar phenomena have been widely investigated in the regime of discriminative models such as Taskonomy [66], they are largely under-explored within a synthesis model. Moreover, we explore two applications of SS-NeRF. We show that the examples synthesized by SS-NeRF (RGB images paired with scene properties) can be used effectively as augmented data for improving the corresponding downstream discriminative tasks. In addition, we show that, because of its learned underlying semantic and geometric scene representations, SS-NeRF can work as an auto-labeller to refine the pseudo-labels produced by state-of-the-art discriminative models.

Our contributions are four-fold: **(1)** We propose a novel solution SS-NeRF to scene understanding from the perspective of learning a synthesis model. To the best of our knowledge, SS-NeRF is *the first work* that extends NeRF to simultaneously rendering photo-realistic novel-view images and *various* corresponding scene properties. **(2)** We instantiate SS-NeRF with five popular scene properties, including semantic labels, surface normal, shading, keypoints, and edges. Intriguingly, as a *versatile neural scene representation*, SS-NeRF outperforms a hybrid strategy that trains NeRF (for rendering images) and task-specific discriminative models (for predicting scene proprieties) separately. **(3)** We show that our SS-NeRF framework is a powerful tool for *bridging generative learning and discriminative learning*, bringing new insight into the investigation of relationships among different scene properties via multi-task learning and knowledge transfer within a synthesis paradigm. **(4)** We further demonstrate that SS-NeRF can benefit a variety of problems, such as facilitating downstream tasks as ways of data augmentation and serving as auto-labeller for data creation.

2. Related Work

Novel-View Synthesis aims to generate a target image with an arbitrary camera pose from one or few given source images [54]. Generative Adversarial Networks [19] (GANs)-based models have shown promising results for synthesizing photo-realistic images of novel views [3, 9, 18, 28, 41, 68]. Though some work also investigates explicitly modeling geometrical properties [7, 20] or introducing 3D shape representations as inductive bias [24, 59, 71], these models still cannot learn implicit 3D representations.

Implicit Scene Representation encodes scenes into feature vectors for novel-view synthesis. Combining the implicit neural model and the volume rendering technology, Neural Radiance Field (NeRF) [37] achieves impressive performance in novel-view synthesis of complicated scenes. It learns an implicit geometric and semantic representation of scenes with perceptron networks and synthesizes views by querying along camera rays with classic volume rendering techniques. Some follow-up work further improves the generalization capability [1, 21, 50, 64], compositionality [22, 42, 44, 67], and efficiency of inference [12, 17, 33, 47]. Inductive biases, such as depth and multi-view consistency, are also introduced to facilitate NeRF-style architectures [43, 55, 57]. Furthermore, the volume rendering technique and underlying semantic and geometric scene representations are also applied to benefit other model structures [30, 34, 49].

While most of the NeRF-based work still focuses on the RGB synthesis task, some explorations have been made to extend NeRF from RGB synthesis to other scene properties. For example, the surface of objects is learned together with the color and density [43], leading to efficient and effective rendering. Geometry representation and reconstruction in neural volume are improved by modeling the surface density [63]. Semantic-NeRF [69] also extends the NeRF-style architecture to semantic annotations, which can be viewed as a *special instance* of our framework, and explores several valuable applications. Different from such work, SS-NeRF scales from RGB synthesis to other pixel-level scene properties, from individual to *multiple* properties, with a shared semantic and geometric scene representation.

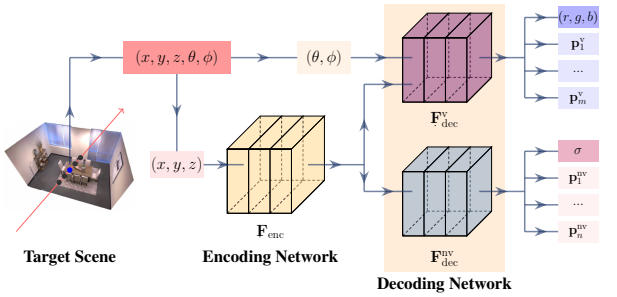

Figure 2: SS-NeRF architecture. The model takes the 3D coordinates and view directions as input and is able to synthesize different paired scene properties. SS-NeRF uses a shared scene encoding network $\mathbf{F}_{\mathrm{enc}}$ to conduct the 3D positional embedding, followed by two separate decoding networks $\mathbf{F}^{\mathrm{v}}_{\mathrm{dec}}$ and $\mathbf{F}^{\mathrm{nv}}_{\mathrm{dec}}$ which produce scene property predictions. $\mathbf{F}^{\mathrm{v}}_{\mathrm{dec}}$ considers the view input, while $\mathbf{F}^{\mathrm{nv}}_{\mathrm{dec}}$ does not.

Recent methods for **Scene Understanding** have gained impressive performance in semantic segmentation [16,23,32, 46], object detection [6, 27, 70], 3D and visual reasoning [4, 10, 26, 35, 45, 60], *etc*. Despite great achievements, few of them focus on understanding scenes from a synthesis model perspective. In comparison, SS-NeRF considers an implicit representation of 3D shape and scene properties, allowing for knowledge transfer and feature sharing across different tasks and thus capturing the underlying image generation mechanism for more comprehensive scene understanding than being done within individual tasks.

Multi-task Learning aims to jointly solve different tasks through leveraging shared knowledge from related tasks [11]. Recent work mainly uses either soft parameter sharing [38, 61] or hard parameter sharing [14, 31] strategies [48]. Beyond solving multi-task learning, the task relationships among different tasks have also been studied. *Taskonomy* and the follow-up work [2, 51, 53, 65, 66] extensively exploit the task relationships to gain the best performance. Compared with prior work, SS-NeRF, as a synthesis model, can also be scaled to solve multiple visual tasks jointly, and further investigate task relationships.

3. Methodology

Fig. 2 illustrates the architecture of our proposed SS-NeRF framework (Scene-Property Synthesis with NeRF). In this section, we first introduce the basic concept of Neural Radiance Fields, followed by the problem setting and innovation of SS-NeRF. Finally, we describe our SS-NeRF design in detail and instantiate it with five representative tasks in the context of scene understanding.

3.1. Neural Radiance Fields

Given a 3D point $\mathbf{x} = (x, y, z)$ and a view direction $\mathbf{d} = (\theta, \phi)$, NeRF [37] learns an implicit scene representation f to map the 5D input to an RGB color $\mathbf{c} = (r, g, b)$ and volume density σ: $f(\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$.

Then NeRF calculates the single pixel color value by tracking and sampling the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, which is emitted from the center $\mathbf{o}$ of the camera plane in the direction $\mathbf{d}$. Specifically, it randomly samples M quadrature points $\{t_m\}_{m=1}^{M}$ with color $\mathbf{c}(t_m)$ and density $\sigma(t_m)$ between the near boundary t_n and far boundary t_f. Then the approximated color of that pixel is given by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{m=1}^{M} \hat{T}(t_m)\alpha(\delta_m \sigma(t_m))\mathbf{c}(t_m), \tag{1}$$

where δ_m is the distance between the two consecutive sample points ($\delta_m = \|t_{m+1} - t_m\|$), $\alpha(d) = 1 - \exp(-d)$, and

$$\hat{T}(t_m) = \exp\left(-\sum_{j=1}^{m-1} \delta_j \sigma(t_j)\right) \tag{2}$$

denotes the accumulated transmittance.

3.2. Innovation and Problem Setting

Innovation: NeRF learns an implicit geometric scene representation with perceptron networks. Our key insight is that this kind of geometry-aware representation can be applicable to and useful for not only RGB color, but also other scene properties, since it is internally shareable. Moreover, such representation addresses limitations of both *discriminative* models (generalization to novel views) and *GAN-based generative* models (generalization from image synthesis to other tasks) coherently. It thus provides a novel synthesis perspective for scene understanding and introduces new potential for a wide range of applications.

Problem Setting: We generalize the basic NeRF setting from single RGB synthesis to rendering additional *pixel-wise scene properties* (*e.g*., semantic labels, edges, surface normal, *etc*.). Specifically, for a certain scene property P_i, we aim to learn a function f_i to estimate its values $\mathbf{p}_i$ for each 3D location and view direction: $f_i(\mathbf{x}, \mathbf{d}) \mapsto \mathbf{p}_i$.

Moreover, since the implicit function encodes the geometry, shape, and texture information of the scene, which are shareable across different property prediction tasks, we argue that different properties can be learned together with shared knowledge. Thus, we further formulate the "Scene-Property Synthesis" problem as follows: Given a collection of K scene properties $\mathcal{P} = \{P_k\}_{k=1}^{K}$, we aim to build a representation function f that can map the 3D coordinates and the view directions to the corresponding property values $f(\mathbf{x}, \mathbf{d}) \mapsto \{\mathbf{p}_k\}_{k=1}^{K}$.

3.3. SS-NeRF

Model Architecture: To solve this novel problem, we propose SS-NeRF, whose model architecture is shown in Fig. 2. Note that, while in principle our framework is applicable to more powerful NeRF variants for improved performance, here we focus on the basic NeRF model [37], showing the effectiveness and generalizability of SS-NeRF *without other advanced components and design choices*. Concretely, the whole model learns to map the 5D vector (3D

coordinates and 2D view directions) to the corresponding scene properties; then we render the scene property "images" with the volume rendering technique used by [37].

We first adopt a shared positional encoder $\mathbf{F}_{\text{enc}}$ to build feature embeddings $e_{\mathbf{x}}$ for the 3D coordinates (x, y, z):

$$e_{\mathbf{x}} = \mathbf{F}_{\text{enc}}(x, y, z). \tag{3}$$

Some scene properties (*e.g.*, semantic labels) are not sensitive to the view direction, so that the view input is redundant. Therefore, we adopt two types of decoding networks $\mathbf{F}^{\text{v}}_{\text{dec}}$ and $\mathbf{F}^{\text{nv}}_{\text{dec}}$ inspired by [69]. $\mathbf{F}^{\text{v}}_{\text{dec}}$ takes the additional view input $\mathbf{d} = (\theta, \phi)$ together with the encoded coordinates to make the predictions for property P^{v}_i, while $\mathbf{F}^{\text{nv}}_{\text{dec}}$ predicts scene property P^{nv}_j directly with the encoded coordinates:

$$\hat{\mathbf{p}}^{\text{v}}_i = \mathbf{F}^{\text{v}}_{\text{dec}}(e_{\mathbf{x}}, \theta, \phi); \quad \hat{\mathbf{p}}^{\text{nv}}_j = \mathbf{F}^{\text{nv}}_{\text{dec}}(e_{\mathbf{x}}). \tag{4}$$

In practice, in our preliminary experiment, we tried these two modeling strategies for each scene property and adopted the one that works better in all the following experiments. We have also validated the necessity of this two-branch model design with ablations in Sec. 4.3.

The simplest working scenario of SS-NeRF is to predict a single scene property. However, by adding more decoding branches, the proposed model is able to predict multiple properties, leading to the generalization from a single task to multiple tasks. In Sec. 4.4, we discuss the application for multi-task learning with SS-NeRF. Notice that the density σ is always required to do the volume rendering for either single property or multiple properties, and the color is the most informative scene property. So we treat them as the fixed outputs for our SS-NeRF model and add other properties upon this basic model.

Instantiation and Optimization of SS-NeRF: We instantiate SS-NeRF with five representative scene properties that are important in practice [51, 66], together with the color image synthesis. These properties are: **Semantic Labels (SL), Surface Normal (SN), Shading (SH), Keypoints (KP), and Edges (ED)**. We adopt $\mathbf{F}^{\text{v}}_{\text{dec}}$ for SH, KP, and ED; and $\mathbf{F}^{\text{nv}}_{\text{dec}}$ for SL and SN.

During the optimization of SS-NeRF, we adopt the hierarchical volume sampling strategy proposed by [37]. That is, we first randomly pick some "coarse" sample points and then produce a more informed sampling of "fine" points that are biased towards the relevant parts of the volume. We also use task-specific objectives for these different properties. For the color image synthesis, we adopt the mean square error (MSE):

$$\mathcal{L}_{\text{rgb}} = \mathcal{L}_{\text{MSE}} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\|\hat{\mathbf{p}}_c(\mathbf{r}) - \mathbf{p}(\mathbf{r})\|^2_2 + \|\hat{\mathbf{p}}_f(\mathbf{r}) - \mathbf{p}(\mathbf{r})\|^2_2 \right], \tag{5}$$

where $\mathbf{p}(\mathbf{r}), \hat{\mathbf{p}}_c(\mathbf{r}), \hat{\mathbf{p}}_f(\mathbf{r})$ are the ground-truth, coarse volume prediction, and fine volume prediction for property P, respectively. $\mathcal{R}$ is the set of rays $\mathbf{r}$ in each batch. The MSE loss is also used for the surface normal prediction.

For semantic label prediction, we use the cross entropy loss function:

$$\mathcal{L}_{\text{seg}} = -\sum_{\mathbf{r}\in\mathcal{R}} \left[\sum_{l=1}^{L} s^l(\mathbf{r}) \log \hat{s}^l_c(\mathbf{r}) + \sum_{l=1}^{L} s^l(\mathbf{r}) \log \hat{s}^l_f(\mathbf{r}) \right], \tag{6}$$

where $s^l, \hat{s}^l_c, \hat{s}^l_f$ are the ground-truth, coarse volume prediction, and fine volume prediction of multi-class semantic probability of class l, respectively. Coarse and fine predictions $\hat{s}^l_c, \hat{s}^l_f$ are processed by a softmax layer after volume rendering. For shading, keypoints, and edges, we adopt the $\mathcal{L}_1$ loss:

$$\mathcal{L}_{\text{ABSE}} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\|\hat{\mathbf{p}}_c(\mathbf{r}) - \mathbf{p}(\mathbf{r})\|_1 + \|\hat{\mathbf{p}}_f(\mathbf{r}) - \mathbf{p}(\mathbf{r})\|_1 \right]. \tag{7}$$

The final loss is the weighted sum of photo-metric loss and the standard loss of the specific task as:

$$\mathcal{L}_{\text{whole}} = \mathcal{L}_{\text{rgb}} + \sum_{P_i\in\mathcal{P}} \lambda_{P_i} \mathcal{L}_{P_i}, \tag{8}$$

where $\mathcal{P} = \{P_{\text{SL}}, P_{\text{SN}}, P_{\text{SH}}, P_{\text{KP}}, P_{\text{ED}}\}$ is the set of properties, and λ_{P_i} is the corresponding weight.

Modeling of Surface Normal: Among all the five scene properties, the surface normal is a special one that is of a vector form, whose projection in the image depends on the camera pose. To better model this property, we use $\mathbf{F}^{\text{nv}}_{\text{dec}}$ as the decoding network but introduce an additional input of encoded camera pose to directly synthesize the encoded normal with the volume rendering technique.

4. Experimental Evaluation

In this section, we evaluate SS-NeRF. We start with the experimental setting in Sec. 4.1, followed by the quantitative and qualitative evaluation for all the five scene properties (Sec. 4.2). In Sec. 4.3, we ablate the model performance without the color image synthesis branch and different decoding networks. We then make further explorations for SS-NeRF, including knowledge transfer and task relationships, data augmentation for downstream discriminative tasks, and also real-world auto-labeller applications (Sec. 4.4). Finally, we discuss the limitations and future work in Sec. 4.5.

4.1. Experimental Setting

Datasets: We first conduct extensive experiments on the commonly-used Replica [52] dataset. Replica is a high-quality synthetic scene dataset containing photo-realistic 3D models for 18 scenes in total. Following [69], we conduct experiments on four scenes and each scene contains 50 frames at resolution 640×480. We have also validated the robustness of our model on the BlendedMVS dataset [62] and investigated the application of SS-NeRF on complex real-world scenes – Local Light Field Fusion (LLFF) collected by [36, 37]. The image resolutions of these two datasets are 768×576 and 4032×3024, respectively. We follow the same processing as NeRF [37] for LLFF.

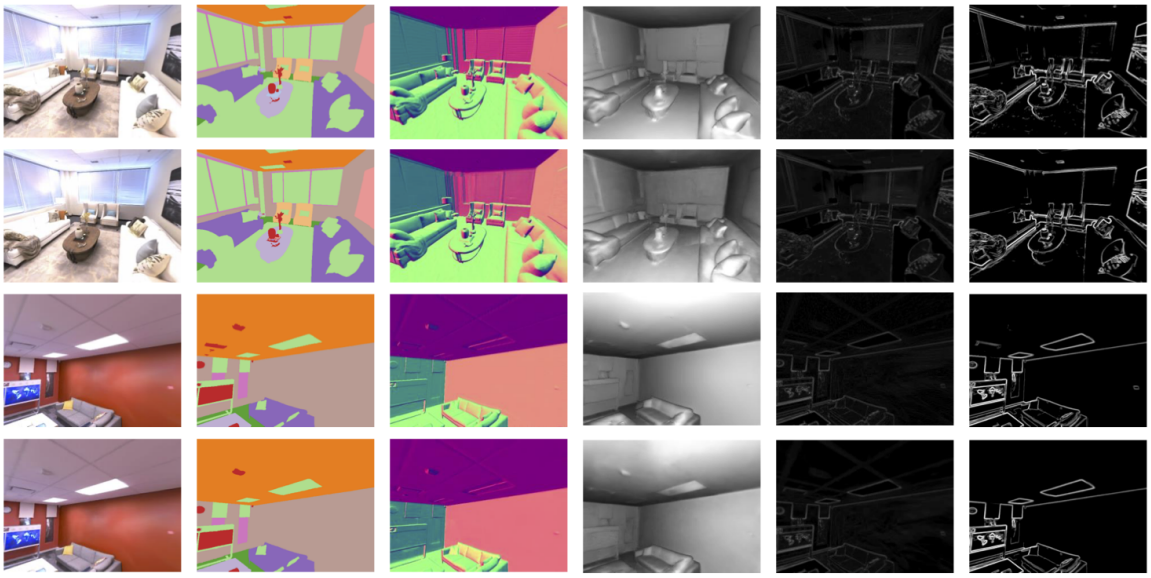

Figure 3: Qualitative results of two representative testing views on Replica. **Top row:** ground-truth; **Bottom row:** our synthesis results. The synthesized RGB images are from the SL task. SS-NeRF is able to render *realistic and matched* RGB images and other properties.

Scene	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)
Office_3	0.9345	0.0355	0.0423	0.0038	0.0155
Office_4	0.9162	0.0383	0.0503	0.0035	0.0150
Room_0	0.9707	0.0323	0.0293	0.0039	0.0209
Room_1	0.8757	0.0520	0.0495	0.0038	0.0202
Avg. (Ours)	**0.9243**	**0.0395**	**0.0429**	**0.0038**	**0.0179**
Avg. (Heuristic)	0.8580	0.0424	0.0451	0.0059	0.0457
Avg. (Hybrid)	0.7360	0.0593	0.0673	0.0055	0.0406

Table 1: Performance of SS-NeRF on individual scene properties. SL: Semantic Labels; SN: Surface Normal; SH: Shading; KP: Keypoint; ED: Edge. SS-NeRF reaches high quantitative scores for all the tasks, and also outperforms the two baselines, indicating that it is able to render accurate scene properties similar to the ground-truth.

Target Properties: Following the observation in [51], we focus on five important scene properties other than the RGB color in our experiments: Semantic Labels (SL), Surface Normal (SN), Shading (SH), Keypoint (KP), and Edge (ED). For Replica, we map the original 88-way semantic classes to the commonly-used NYUv2-13 [15, 40] format.

Scene Annotations: We render the missing annotations by ourselves. The surface normal is derived from the depth by $SN(x, y, z) = (-\frac{dx}{dz}, -\frac{dy}{dz}, 1)$, where (x, y, z) are the 3D coordinates and $\frac{dx}{dz}$, $\frac{dy}{dz}$ are the gradients of z with respect to x and y, respectively. Edges are rendered by a Canny [8] detector; Keypoints are derived from SURF [5]; Shadings are rendered by a pre-trained model, XTConsistency [65].

Implementation Details: Consistent with NeRF [37], we optimize our model for each scene separately. We set $\lambda_{\text{SN}} = 1$, $\lambda_{\text{SL}} = 0.04$, $\lambda_{\text{SH}} = 0.1$, $\lambda_{\text{KP}} = 2$, and $\lambda_{\text{ED}} = 0.4$ via cross-validation. We use the Adam optimizer [29] with an initial learning rate of 5×10^{-4} and set $\beta_1 = 0.9, \beta_2 = 0.999$. We train our model for 200k iterations on each scene, taking about 9 hours on a single NVIDIA RTX 2080 Ti GPU.

Setting	SL (↑)	SH (↓)	KP (↓)	ED (↓)
$\mathbf{F}^{\text{v}}_{\text{dec}}$	0.9173	**0.0429**	**0.0038**	**0.0179**
$\mathbf{F}^{\text{nv}}_{\text{dec}}$	**0.9243**	0.0745	0.0039	0.0211

Table 2: Ablation of average results on different modeling for the four scene properties on 4 scenes in Replica. The view input is critical for SH, KP, and ED, but is redundant for SL.

Evaluation Metrics: We use mean Intersection-over-Union (mIoU) to evaluate the semantic segmentation and $\mathcal{L}_1$ error to measure the performance of other tasks.

4.2. Performance on Tasks Beyond RGB

We first build SS-NeRF for each individual scene property and evaluate them on Replica. We report the quantitative results in Table 1. Note that the main objective of this paper is to show that, with SS-NeRF, it is able to *synthesize* different scene properties paired with the rendered images; therefore, there is no existing work as baselines for a more comprehensive comparison. While there has been a large body of work on training discriminative models to predict the scene properties for *real* images, it is difficult to make an apple-to-apple comparison between these discriminative models and our synthesis model. Conceptually, the synthesis models can in principle produce infinite paired samples, while the discriminative models are constrained by the given data.

However, to have a better understanding, we compare the model performance with one heuristic baseline and one hybrid baseline. **Heuristic Baseline (Heuristic)** estimates the annotations of the test view by finding the nearest view in the training set, and then mapping the source labels directly to

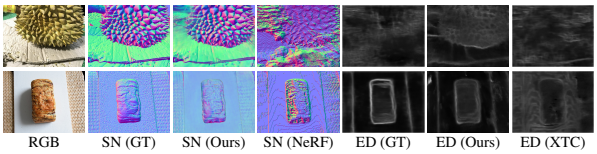

Figure 4: Representative results on blendedMVS. SN (NeRF) is the normal derived from NeRF's depth; ED (XTC) is the edge predicted from a well-trained model taking NeRF's normal as input. Our model outperforms both methods, indicating the capability and robustness of SS-NeRF.

the target view with perspective projection. **Hybrid Baseline (Hybrid)** trains a synthesis model (NeRF) and task-specific discriminative models separately. For novel test views, we first generate the color image corresponding to that pose, and then predict the annotations with the well-trained annotators. We adopt standard Taskonomy encoding-decoding architectures [66] for Hybrid. We report the averaged results for all the scenes in Table 1.

From Table 1, we have the following observations: (1) SS-NeRF reaches a high performance for all the five tasks, indicating that our model can well capture the original distribution of all the scene properties; (2) SS-NeRF outperforms the heuristic baseline for all the tasks, which verifies the accurate label quality generated by our SS-NeRF model; (3) SS-NeRF also outperforms the hybrid baseline for all the tasks, showing that it is non-trivial to synthesize paired color images and other scene properties, and that the shared semantic and geometric scene representation is critical for synthesizing different scene properties.

We also visualize our rendered scene properties and compare with the corresponding ground-truth in Fig. 3. All the images manifest the good novel-view synthesis results from our SS-NeRF for additional scene properties beyond RGB. Moreover, we have also conducted experiments on the real-world BlendedMVS dataset [62] to verify the robustness. Two samples for SN and ED are shown in Fig. 4. For SN, we compare with the normal derived by the NeRF's depth; for ED, we compare with an even stronger hybrid baseline, XTConsistency [65] that contains a powerful backbone and is pre-trained on Taskonomy [66], working on the synthesized images by NeRF. Our model has obviously better visualizations for the challenging "durian" scenario on both tasks. For the simple "bread" scenario, we capture more details and our prediction for ED is closer to the ground-truth.

4.3. Ablation Study

Modeling with the two Decoders: In Sec. 3, we propose two branches for different scene properties. For each scene property except SN (special modeling), we choose the one with a better performance. In Table 2, we show the quantitative comparison between the two types of modeling for each scene property. We find that $\mathbf{F}^{\mathrm{v}}_{\mathrm{dec}}$ works better for SH, KP, and ED, but cannot beat $\mathbf{F}^{\mathrm{nv}}_{\mathrm{dec}}$ for SL. This observation is consistent with the intuition: shading, keypoints, and edges vary from different view directions, but semantic labels keep the same. Therefore, the view input is critical for SH, KP, and ED but redundant for SL. This observation also indicates that SS-NeRF indeed learns a geometry-aware representation for the scenes.

Model	PSNR (↑)
NeRF	29.9230
SS-NeRF-SL	**30.2019**
SS-NeRF-SN	29.8111
SS-NeRF-SH	28.1492
SS-NeRF-KP	29.7657
SS-NeRF-ED	28.8192

Property	w/o RGB (Avg.)	w/ RGB (Avg.)
SL (↑)	0.5208	**0.9243**
SN (↓)	0.0440	**0.0395**
SH (↓)	0.0551	**0.0429**
KP (↓)	0.0114	**0.0038**
ED (↓)	0.0560	**0.0179**

Table 3: Ablation study on the RGB color branch. **Left:** averaged PSNR measurement for the basic NeRF and SS-NeRF variants. Other scene properties will not affect the visual quality of the synthesized images. **Right:** performance comparison between the models with or without the RGB branch. RGB supervision is crucial for understanding the scenes and learning other visual properties.

Setting	Office_3	Office_4	Room_0	Room_1
SH	0.0423	0.0503	0.0293	0.0495
SH + SL	0.0417(+)	0.0479(+)	0.0295(-)	0.0432(+)
SH + SN	0.0403(+)	0.0471(+)	0.0303(-)	0.0445(+)
SH + KP	0.0427(-)	0.0478(+)	0.0296(-)	0.0473(+)
SH + ED	0.0422(+)	0.0483(+)	0.0311(-)	0.0501(-)
SH + All	0.0415(+)	0.0481(+)	0.0318(-)	0.0452(+)

Table 4: Model performance with additional tasks for shading. (+) indicates performance increasing, and (-) indicates performance drop. SL consistently benefits the target SH task for nearly all the scenes, indicating a closer relationship between these two tasks.

Modeling for RGB: The RGB color is a fundamental scene property and can facilitate the learning of the other properties. Here we ablate the key role of RGB color in two sets of experiments. First, we measure the averaged quality of the synthesized RGB images with peak signal-to-noise ratio (PSNR) for the basic NeRF model and all the variants of our SS-NeRF. The left part of Table 3 shows that joint training of RGB and other scene properties will not affect the visual quality of the synthesized images. Furthermore, including SL even improves the PSNR of the basic NeRF. Next, we build another variant of SS-NeRF for each scene property that removes the RGB color output (w/o RGB). The averaged performance among all the scenes is shown in the right part of Table 3. Based on the result, we find that RGB supervision is crucial for understanding the scenes and learning other visual properties.

4.4. Further Explorations within SS-NeRF

Multi-task Learning: We instantiate SS-NeRF for each single scene property but it is able to simultaneously learn scene representations and shared knowledge within multiple visual tasks, so as to further benefit individual tasks. Taking SH as an example, we further build five variants under different task settings to conduct multi-task learning and also investigate whether other tasks can benefit from semantic

Settings	Office_3	Office_4	Room_0	Room_1
Limited Views	0.1171	0.0993	0.0685	0.1246
SL $\rightarrow$ SH	0.0915	0.0886	0.0606	0.0982
SN $\rightarrow$ SH	0.0917	0.0911	0.0606	0.1002
KP $\rightarrow$ SH	0.0893	0.0864	0.0607	0.1016
ED $\rightarrow$ SH	0.0920	0.0864	0.0585	0.0965

Table 5: Model performance with transfer learning. With the learned shareable knowledge from other scene properties, the transferred model consistently achieves better performance, indicating generalizability of SS-NeRF.

segmentation in the framework of SS-NeRF. We first introduce the other four properties to be jointly trained with SH (denoted as SH + "additional property"), and also build a variant that is trained for all the five properties (SH + All).

We show the results in Table 4. We have the following observations: (1) SL consistently benefits the target SH task for all scenes except "Room_0," but the gap is marginal, indicating a closer relationship between the two tasks. It may be because the semantics label implicitly contains the texture and geometry information of the scene, which makes the model better estimate the shading. (2) Jointly training with all the tasks outperforms the single task model in three out of four scenes, indicating the general benefit of the knowledge from other scene properties. (3) Model performance also varies in different scenes, indicating that the task relationships might also rely on the scene structures, and the relationship among tasks might not be stationary for generative models. Interestingly, these observations are also consistent with those for discriminative models [51, 66].

Knowledge Transfer: In additional to investigating multi-task learning, we explore the generalization of the learned scene representations by conducting transfer learning. Still taking SH as the target scene property, we first train our model with another source property and transfer the knowledge learned by the source to the target SH through initializing the learned encoding network $\mathbf{F}_{\text{enc}}$. Different from previous experiments, here we focus on the typical transfer learning setting with limited data (6 training views) for the target property. The results are shown in Table 5, for which "Limited Views" is the baseline without knowing any prior knowledge. We can find that with the learned shareable knowledge from other scene properties, the transferred model can consistently achieve better performance, indicating the effective generalization of the SS-NeRF framework.

Data Augmentation for Multi-task Learning: Given that we can render photo-realistic images and their corresponding scene property annotations, one natural, interesting question arises: How can we make use of these paired synthesized data? Inspired by [2, 13], we design the following experiment. We adopt a task network (*i.e.*, a standard discriminative model) to evaluate each task, and we train this model under four data settings: (1) ground-truth (GT); (2) paired RGB images and corresponding annotations generated by SS-NeRF (SS-NeRF); (3) ground-truth and augmented data generated by our model (GT+SS-NeRF); (4) ground-truth and augmented novel-view data synthesized by SS-NeRF (GT+SS-NeRF-N). For the GT+SS-NeRF data setting, we generate paired data with the same poses as GT; for the last setting, we generate data from novel views (averaged view from adjunct views in the training set). For the task network, we adopt a standard Taskonomy encoding-decoding architecture [66]. Different from the main experiment, we combine all the data from the four scenes together for evaluation. We train all the models for 200 epochs.

Data Setting	SL(↑)	SN(↓)	SH(↓)	KP(↓)	ED(↓)
GT	0.5805	0.0394	0.0610	0.0051	0.0229
SS-NeRF	0.5575	0.0434	0.0594	0.0048	0.0268
GT + SS-NeRF	**0.6178**	0.0394	0.0552	0.0048	0.0224
GT + SS-NeRF-N	0.5929	**0.0390**	**0.0531**	**0.0041**	**0.0206**

Table 6: Comparison of the four data settings. GT: paired ground-truth data; SS-NeRF: paired synthesized data; GT+SS-NeRF: GT data and augmented data rendered by SS-NeRF (same pose); GT+SS-NeRF-N: GT data and augmented data rendered by SS-NeRF (novel pose). SS-NeRF synthesizes both visually realistic and useful data, so it can be used as an effective way of data augmentation to benefit the learning of other visual tasks.

The results are shown in Table 6. We find: (1) GT and SS-NeRF have comparable performance, and SS-NeRF even outperforms GT in SH and KP, indicating the good quality of the data generated by SS-NeRF. (2) For all the five scene properties, including the augmented data, even from the same pose, can bring additional improvements. (3) This improvement further increases for most tasks when we use augmented data from novel views. These results indicate that SS-NeRF can generate both visually realistic and useful data, making it attractive to be applied to benefit the learning of visual perception tasks.

Auto-Labelling for Real-World Scenes: One important application for the multi-task discriminative models is that they work as auto-labeller to annotate the real-world data, after pre-trained on synthetic or academic small-scale datasets. Our SS-NeRF model can be used as auto-labeller as well. Note that, different from discriminative models that directly operate on real images, our SS-NeRF simultaneously renders images and their per-pixel scene property annotations. Considering this difference, we introduce a *two-stage* procedure for leveraging SS-NeRF as an auto-labeller. With a pre-trained discriminative model, we first produce initial ground-truth annotations. Such annotations are not guaranteed to be correct and could be even flawed – *e.g.*, they might be inconsistent across different views. Then, we train SS-NeRF with these weak annotations. Because SS-NeRF can implicitly learn the semantic and geometric scene representation, it can *correct these inconsistencies* during optimization. This refinement as auto-labeller is reminiscent of a denoising task (in [69]), which aims to correct the minor noisy ground-truth by learning from the majority of the accurate labels.

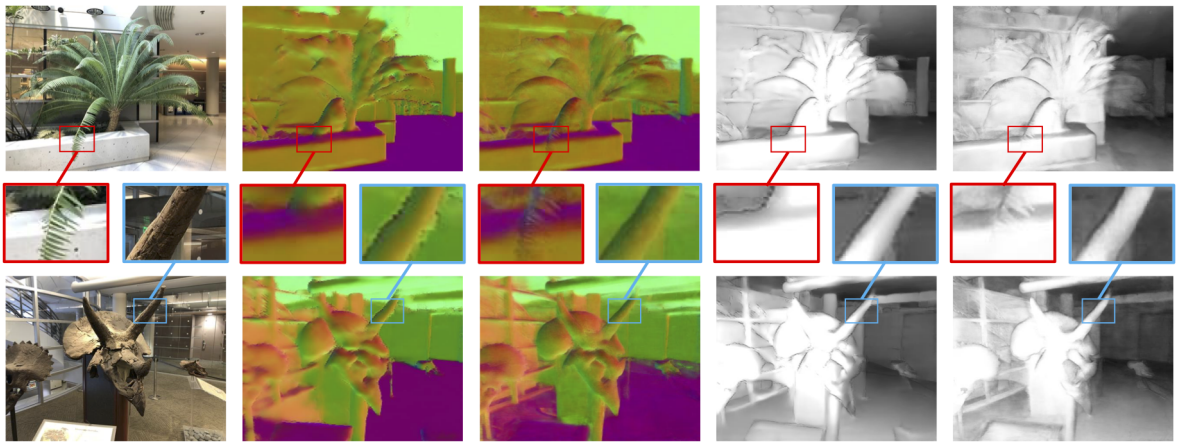

Figure 5: Surface normal and shading predictions with real-world images from the LLFF dataset. We use pre-trained annotators to obtain the initial labels which are noisy and flawed, and we retrain SS-NeRF with these labels. SS-NeRF can refine these flawed annotations and restore more details by joint modeling and understanding of the scenes.

However, the auto-labelling task is more challenging, since there is no guarantee that the majority of the annotations is accurate and the model has to detect and refine the correct labels based on the underlying 3D geometry.

Based on this insight, we move to a real-world dataset without annotations – the LLFF dataset [36]. We also use a pre-trained annotator [65] to generate weak annotations for this dataset (2nd and 4th columns in Fig. 5). Due to the data distribution gap between the LLFF and Taskonomy datasets, the quality of these annotations is quite poor; *e.g.*, for the surface normal, there are sharp faults in the object boundaries. Then we train SS-NeRF with these flawed annotations and we show the results for surface normal and shading on two scenes of the LLFF dataset in Fig. 5. It is clear to see that our SS-NeRF produces smoother results, contains more details, and reflects better 3D structures of the scene. We argue that the refinement comes from the joint modeling and understanding of the scenes, inherent within the SS-NeRF framework, showing the capability of our model in scene understanding. In addition, this general idea of auto-labelling and refinement can be in principle applied to other real-world data and jointly work with other discriminative models.

4.5. Limitations and Future Work

There are two major limitations for our SS-NeRF model: (1) SS-NeRF builds upon the original NeRF model, which is scene-dependent, making it hard to transfer the learned knowledge from one scene to another; (2) SS-NeRF requires accurate and dense pose annotations to learn scene representations, which might not be accessible for all the datasets (*e.g.*, Taskonomy [66]). Notice that these limitations are essentially from the original NeRF model and some follow-up work has provided promising solutions [1, 21, 50, 56, 64]. Similar techniques can be introduced to our SS-NeRF framework to further enhance the model capability.

Our work provides a first versatile representation for scene property synthesis based on neural radiance fields. The high-level motivation is that the underlying semantic and geometric scene representation from NeRF facilitates the knowledge sharing across different tasks, therefore enabling it to extend from color image synthesis to other scene properties. Investigating similar strategies for other formats of scene representations, such as point clouds [58] and meshes [25], can also be promising directions for future research.

5. Conclusion

This work shows that a comprehensive scene representation with implicitly encoded 3D geometry and semantic structure, powered by the NeRF-style architecture, can be useful for not only RGB image synthesis tasks, but also various visual tasks. Inspired by this, we propose a unified framework SS-NeRF that allows knowledge and representation sharing across different tasks. This novel strategy of solving visual perception problems with a synthesis model provides a different perspective for multi-task learning, which is normally tackled in the context of discriminative models. We further show some interesting observations and promising applications within this synthesis framework.

Acknowledgement: We thank Jun-Yan Zhu, Pavel Tokmakov, and Robert Collins for their valuable comments. This work was supported in part by NSF Grant 2106825, Toyota Research Institute, NIFA award 2020-67021-32799, the Jump ARCHES endowment through the Health Care Engineering Systems Center, the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign through the NCSA Fellows program, and the IBM-Illinois Discovery Accelerator Institute.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. TöRF: Time-of-flight radiance fields for dynamic scene view synthesis. In *NeurIPS*, 2021. 2, 8

[2] Zhipeng Bao, Martial Hebert, and Yu-Xiong Wang. Generative modeling for multi-task visual learning. In *ICML*, 2022. 1, 3, 7

[3] Zhipeng Bao, Yu-Xiong Wang, and Martial Hebert. Bowtie networks: Generative modeling for joint few-shot recognition and novel-view synthesis. In *ICLR*, 2021. 2

[4] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 3

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 5

[6] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *ECCV*, 2020. 3

[7] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 2

[8] John Canny. A computational approach to edge detection. *PAMI*, 1986. 5

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016. 2

[10] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, 2018. 3

[11] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 3

[12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. 1, 2

[13] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In *ECCV*, 2020. 7

[14] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 3

[15] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 5

[16] Rui Fan, Hengli Wang, Peide Cai, and Ming Liu. Sne-roadseg: Incorporating surface normal information into semantic segmentation for accurate freespace detection. In *ECCV*, 2020. 3

[17] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200fps. In *ICCV*, 2021. 1, 2

[18] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. GANalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019. 2

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[20] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019. 2

[21] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2, 8

[22] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2

[23] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *ICCV*, 2019. 1, 3

[24] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2D data to learn textured 3D mesh generation. In *CVPR*, 2020. 2

[25] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3D sheet for view synthesis from a single image. In *ICCV*, 2021. 8

[26] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017. 3

[27] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021. 1, 3

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 2

[29] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[30] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *3DV*, 2020. 2

[31] Iasonas Kokkinos. UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 3

[32] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *ECCV*, 2020. 3

[33] David B Lindell, Julien NP Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 1, 2

[34] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. GNeRF: GAN-based neural radiance field without posed camera. In *ICCV*, 2021. 2

[35] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019. 3

[36] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *TOG*, 38(4), 2019. 4, 8

[37] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 3, 4, 5

[38] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 3

[39] Muzammal Naseer, Salman Khan, and Fatih Porikli. Indoor scene understanding in 2.5/3D for autonomous agents: A survey. *IEEE Access*, 2018. 1

[40] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5

[41] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In *ICCV*, 2019. 2

[42] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 1, 2

[43] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021. 2

[44] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021. 2

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3

[46] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. In *ECCV*, 2020. 3

[47] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021. 1, 2

[48] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 3

[49] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. *CVPR*, 2022. 2

[50] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. *NeurIPS*, 2020. 1, 2, 8

[51] Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020. 3, 4, 5, 7

[52] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 4

[53] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. In *NeurIPS*, 2020. 3

[54] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[55] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 2

[56] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 8

[57] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2

[58] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 8

[59] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NeurIPS*, 2016. 2

[60] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 3

[61] Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. In *ICLR*, 2017. 3

[62] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020. 4, 6

[63] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 2021. 2

[64] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 8

[65] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020. 3, 5, 6, 8

[66] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 2, 3, 4, 6, 7, 8

[67] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[68] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. In *ECCV*, 2018. 2

[69] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 2, 4, 7

[70] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *ECCV*, 2020. 3

[71] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3D representation. In *NeurIPS*, 2018. 2